International Journal of Computer Science Engineering
and Information Technology Research (IJCSEITR)
ISSN(P): 2249-6831; ISSN(E): 2249-7943
Vol. 3, Issue 5, Dec 2013, 43-46
© TJPRC Pvt. Ltd.

TRANS STELLAR
•Journal Publications •Research Consultancy

BISECTING COLLINEAR CLUSTERING ALGORITHM

TERENCE JOHNSON

Department of Computer Engineering, Agnel Institute of Technology and Design, Goa, India

ABSTRACT

The motivation for the Bisecting Collinear Clustering Algorithm comes from the Bisecting K-means Algorithm. To obtain T clusters, split the set of all points into two clusters, select one of these clusters to split (Each time Bisect the selected cluster using the Collinear Clustering Algorithm), and so on, until T clusters have been produced. Clustering, using the classical K-Means method results in obtaining final fixed points which we call the final unchanging means around which all other points in the dataset get clustered. This suggests that if we identify points in any dataset which represent the final unchanging means, then the task of clustering reduces to just assigning the remaining points in the dataset into clusters, which are closest to these final means based on standard distance measures. Taking a cue from the result of the K-Means method, the collinear clustering algorithm groups collinear data points of any dataset into different clusters, depending on which points in the dataset lie at maximum distance from each other. The clusters are formed by finding the minimum Euclidean distance of all points in the dataset and these maximally separated data points.

KEYWORDS: Bisecting Collinear Clustering, Maximal Distance Clustering, Minimum Euclidean Distance

INTRODUCTION

Clustering can be used in various applications like Market Segmentation, grouping of genetic code for medical research and almost all business organizations. Clustering deals with a collection of objects which are similar between them and are dissimilar to objects belonging to other clusters. Clusters are groupings or collection of points such that the maximum distance between all pairs of points in a particular group is as small as possible. Clustering organizes and partitions objects into groups whose members are alike in some way. It is obvious from the above arguments that clustering hinges on the notion of distance. In order to decide whether a set of points can be split into subclusters, with members of a cluster being closer to other members of their cluster than to members of other clusters, we need to say what we mean by "closer to".

Closeness or similarity and distance or dissimilarity can be described by standard distance norms.

THE BISECTING COLLINEAR CLUSTERING ALGORITHM

To obtain T clusters, split the set of all points into two clusters, select one of these clusters to split , each time Bisect the selected cluster using the Collinear Clustering Algorithm [1], and so on, until T clusters have been produced.

Algorithm 1. The Bisecting Collinear Clustering Algorithm

Input: **The number of clusters K and a database containing n objects.**

Output**: A set of K clusters.**

```
    Initialize the list of clusters to contain the cluster consisting of all points.
    repeat
```

```
        Remove the largest cluster from the list of clusters
        {Perform several "trial" bisections of the chosen
           cluster}
        for i = 1 to number of trials do (Bisect the selected cluster using the Collinear Clustering
      Algorithm[1] shown below as Algorithm 2.)  Find $T_{min}$ and $T_{max.}$ from D
                    repeat(For K = N+1)
                          $TX_m = T_{min} + X_m \left[\frac{d(T_{\min}, T_{\max})}{K-1}\right]$
                          If $TX_m \not\subset$ D, then find a value say $T_c$ in the dataset D which is closest to $TX_m$ using
      minimum
                    Euclidean distance measure and assign $T_c$ to $TX_m$
      Until Output K=N+1 clusters.
        end for
        Select the two clusters from the bisection with the highest overall similarity.
        Add these two clusters to the list of clusters.
      until until the list of clusters contains K clusters.
```

Algorithm 2. The Collinear Clustering Algorithm.

Input: **The number of clusters K and a database containing n objects**

Output**: A set of K clusters.**

```
      Initialize the list of clusters to contain the cluster consisting of all points.
      Repeat
        Remove the largest cluster from the list of clusters
        {Perform several "trial" bisections of the chosen
           cluster}
        for i = 1 to number of trials do (Bisect the selected cluster using the Collinear Clustering
      Algorithm[1] shown below as Algorithm 2.)  Find $T_{min}$ and $T_{max.}$ from D
          repeat(For K = N+1)
                 $TX_m = T_{min} + X_m \left[\frac{d(T_{\min}, T_{\max})}{K-1}\right]$
                 If $TX_m \not\subset$ D, then find a value say $T_c$ in the dataset D which is closest to $TX_m$ using
      minimum
             Euclidean distance measure and assign $T_c$ to $TX_m$
```

```
    Until Output K=N+1 clusters.
      end for
      Select the two clusters from the bisection with the highest overall similarity.
      Add these two clusters to the list of clusters.
until until the list of clusters contains K clusters.
```

EXPERIMENTAL RESULTS

For a given one dimensional dataset D = {2, 5, 12, 34, 14, 19, 12, 24, 32, 3, 20, 30, 25} and given clustering requirement as K = 4 clusters, after implementation, the four clusters were successfully found out through the various iterations show below to be:

After Iteration 1

Cluster 1 = {2, 3, 5, 12, 14, 19} and

Cluster 2 = {20, 24, 32, 34}.

After Iteration 2

Cluster 1 = {2, 3, 5}

Cluster 2 = {12, 14, 19} and

Cluster 3 = {20, 24, 32, 34}.

After Iteration 3

Cluster 1 = {2, 3, 5}

Cluster 2 = {12, 14, 19}

Cluster 3 = {20, 24} and

Cluster 4 = {32, 34}.

After Iteration 4

Cluster 1 = {2, 3},

Cluster 2 = {5}

Cluster 3 = {12, 14,19}

Cluster 4 = {20, 24} and

Cluster 5 = {32, 34}.

After Iteration 5

Cluster 1 = {2, 3}

Cluster 2 = {5}

Cluster 3 = {12, 14}

Cluster 4 = {19}

Cluster 5 = {20, 24} and

Cluster 6 = {32, 34}.

After selecting the two clusters from the bisection with the highest overall similarity and adding these two clusters to the list of clusters we get the required 4 clusters as

Cluster 1 = {2, 3, 5}

Cluster 2 = {12, 14}

Cluster 3 = {19, 20, 24} and

Cluster 4 = {32, 34}.

There a number of different ways to choose which cluster to split. We can choose the largest cluster at each step, choose the one with the largest Sum of Squared Error, or use a criterion based on both size and Sum of Squared Error.

CONCLUSIONS

The analysis shows how the Bisecting Collinear Clustering Algorithm finds four clusters in the data by way of five iterations to form the required four clusters. In Iteration 1, two clusters are found. In Iteration 2, the left-most cluster is split and in Iteration 3, the left-most cluster is split. In this way, the required number of clusters are found in five iterations.

REFERENCES

1. Terence Johnson, Jervin Zen Lobo, Collinear Clustering Algorithm, IOSR Journal of Computer Engineering (IOSRJCE) ISSN:2278-0661, ISBN: 2278-8727 Volume 6, Issue 5 (Nov. - Dec. 2012), PP 08-11
2. Pang-Ning Tan, Michael Steinbachand Vipin Kumar, Introduction to data mining (Addison Wesley, 2006)
3. David Hand, Heikki Mannila and Padhraic Smyth, Principles of data mining (Cambridge, MA: MIT Press, 2001)
4. Jiawei Han and Micheline Kamber, Data mining-concepts and techniques (San Francisco CA, USA, Morgan Kaufmann Publishers, 2001)
5. A.K. Jain, R.C. Dubes, Algorithms for clustering data, (Englewood Cliffs, NJ: Prentice-Hall, 1998)
6. M.R. Anderberg, Cluster analysis for application, (Academic Press, New York, 1973)
7. J.A. Hartigan, Clustering Algorithms, (Wiley, New York, 1975)
8. Hand, D.J., Blunt, G., Kelly, M.G. & Adams, N.M. (2000), Data mining for fun and profit, (Statistical Science) 15, 111-131.
9. Fayyad, U., Data Mining and Knowledge Discovery, Editorial, Proc. IEEE, 1:5-10, 1997. W.J. Book, Modelling design and control of flexible manipulator arms: A tutorial review, Proc. 29th IEEE Conf. on Decision and Control, San Francisco, CA, 1990, 500-506
10. Aggarwal, Charu C., Han,Jiawei,Wang, Jianyong, & Yu, Philip S. A framework for clustering evolving data streams, VLDB Endowment, Proceedings of the 29th international conference on very large data bases, Vldb 2003, 81–92